

I Claim:

1. A method for assessing the relative complexity of different options for performing a task, the method comprising the steps of:

5

defining the task as a sequenced set data structure that specifies actions of the task, and sequence information that specifies the order in which particular actions are to be performed;

10

storing recipes available for performing constituent actions of the task as sequenced set data structures that specify subactions of the recipes for the constituent actions, and sequence information that specifies the order in which the subactions are to be performed; and

15

determining complexity measures associated with performing the task using different combinations of recipes for constituent actions of the task, based upon complexity measures of actions specified by respective combinations of available recipes.

20

2. The method as claimed in claim 1, wherein complexity measures for actions are defined in terms of the complexity measures of available recipes for performing the actions, and complexity measures for recipes are defined in terms of the complexity of the subactions of the recipe.

25

3. The method as claimed in claim 1, further comprising the steps of:

determining predetermined complexity measures for basic actions that are not specified by a recipe; and

30

determining specified complexity measures for contracted actions that are performed by a different agent.

4. The method as claimed in claim 1, further comprising the alternating steps of:

updating complexity measures for recipes in relation to actions whose complexity measures are determined; and

5 updating complexity measures for actions in relation to recipes whose complexity measures are updated.

5. The method as claimed in claim 1, wherein the complexity measures associated with a particular action performed by a particular agent is based upon the
10 complexity measures for each of the recipes for that action.

6. The method as claimed in claim 1, wherein the sequencing relation specifies, for pairs of actions, that one specified action is sequenced before another specified action.

15 7. The method as claimed in claim 1, further comprising the step of defining a sequenced set data structure as $S = (A, M)$, in which A is a multi-set element and M is a sequencing relation that specifies an ordered sequence of the elements A in the sequenced set S .

20 8. The method as claim in claim 7, further comprising the step of defining a sequencing relation for the sequenced set data structure S for two elements a_i and a_j of multi-set element A , such that a_i is sequenced before a_j in set A under the relation M .

25 9. The method as claimed in claim 1, further comprising the step of delegating the defined task to a primary agent for execution of the task by at least one of the primary agent and one or more contracting agents.

30 10. The method as claimed in claim 1, further comprising the step of performing the defined task by executing a series of actions for which the global measure of complexity for the task is determined to be a minimum.

11. Computer software recorded on a medium for assessing the relative complexity of different options for performing a task, the computer software comprising:

software code means for defining a task as a sequenced set data structure that specifies actions of the task, and sequence information that specifies the order in which particular actions are to be performed;

software code means for storing recipes available for performing constituent actions of the task as sequenced set data structures that specify subactions of the recipes for constituent actions, and sequence information that specifies the order in which the specified subactions are to be performed; and

software code means for determining complexity measures associated with performing the task using different combinations of recipes for constituent actions of the task, based upon complexity measures of actions specified by respective combinations of available recipes.

12. Computer software recorded on a medium for assessing the relative complexity of different options for performing a task, the computer software comprising software code for performing the following steps:

defining a task as a sequenced set data structure that specifies actions of the task, and sequence information that specifies the order in which particular actions are to be performed;

storing recipes available for performing constituent actions of the task as sequenced set data structures that specify subactions of the recipes for constituent actions, and sequence information that specifies the order in which the specified subactions are to be performed; and

determining complexity measures associated with performing the task using different combinations of recipes for constituent actions of the task, based

upon complexity measures of actions specified by respective combinations of available recipes.

- 5 13. A computer system for assessing the relative complexity of different options for performing a task, the computer system comprising:

means for defining a task as a sequenced set data structure that specifies actions of the task, and sequence information that specifies the order in which particular actions are to be performed;

10 means for storing recipes available for performing constituent actions of the task as sequenced set data structures that specify subactions of the recipes for constituent actions, and sequence information that specifies the order in which the specified subactions are to be performed; and

15 means for determining complexity measures associated with performing the task using different combinations of recipes for constituent actions of the task, based upon complexity measures of actions specified by respective combinations of available recipes.

- 20 14. A computer system for assessing the relative complexity of different options for performing a task, the computer system comprising:

25 a logic element for defining a task as a sequenced set data structure that specifies actions of the task, and sequence information that specifies the order in which particular actions are to be performed;

30 a memory for storing recipes available for performing constituent actions of the task as sequenced set data structures that specify subactions of the recipes for constituent actions, and sequence information that specifies the order in which the specified subactions are to be performed; and

a data processor for determining complexity measures associated with performing the task using different combinations of recipes for constituent actions of the task, based upon complexity measures of actions specified by respective combinations of available recipes.